Location.java – Changes

Timestamp field was changed to “long” type:

**private** **final** **long** tmstmp;

New Constructor was added:

**public** Location(String title, **long** tmstmp, String text, **double** latitude, **double** longitude) {

**this**(-1, title, tmstmp, text, latitude, longitude);

}

It’s used for return while getting the information by ID

Response.java – Used for response from server

It contains:

**private** **int** status = -1; //status of the reques 0-error, 1-ok

**private** **int** type = -1; //type if status = 1 => 1-insert, 2-get, 3-delete

**private** Location location = **null**; //if status = 1 => if type = 1 – contains new ID

//if type = 2 – contains all info from server

//if type = 3 – contains ID that was deleted

**private** String explanation = **null**; //explanation of the error if status = 0

**private** String additionalinfo = **null**; //additional info about the operation (for debugging)

To use the functionality with server you need the HttpRequestPerformer class:

1. Inserting:

//form the Location object

//for date use function calendar.getTime().getTime(); it returns the date in long type

//to get the Date back from long, you should use DateFormat class

//where the Calendar calendar = Calendar.*getInstance*();

location = **new** Location(0, title, date, text, latitude, longitude);

response = HttpRequestPerformer.*postInfo*(location);

**if**(response == **null**) {

Log.*w*("Response", "Empty response");//for debugging

**return** **false**;

}

**else** {

tvStatus.setText(String.*valueOf*(response.getStatus()));

//new ID you will get here

//also you should check if response.getLocation() is not null

etId.setText(String.*valueOf*(response.getLocation().getId()));

}

1. Getting:

//Form a Location object just with ID

location = **new** Location(Integer.*parseInt*(etId.getText().toString()));

response = HttpRequestPerformer.*getInfo*(location);

**if**(response == **null**) {

Log.*w*("Response", "Empty response");

**return** **false**;

}

**else** {

//then you will get the Location with: Title, Text, Date (long), Latitude, //Longtitude

tvStatus.setText(String.*valueOf*(response.getStatus()));

etTitle.setText(response.getLocation().getTitle());

etText.setText(response.getLocation().getText());

etLat.setText(String.*valueOf*(response.getLocation().getLatitude()));

etLong.setText(String.*valueOf*(response.getLocation().getLongitude()));

}

1. Deleting:

//Form the Location just with ID

location = **new** Location(Integer.*parseInt*(etId.getText().toString()));

response = HttpRequestPerformer.*deleteInfo*(location);

**if**(response == **null**) {

Log.*w*("Response", "Empty response");

**return** **false**;

}

**else** {

//here you will get the status (also you can use the deleted ID from //response.getLocation().getId();

tvStatus.setText(String.*valueOf*(response.getStatus()));

}

UML Diagram I don’t know UML very good. So I uploaded my UML diagram (mainUml.uml) to the Travel Diary SVN.

I’ve uploaded the working Application (for testing of my part) to another branch.

You need to take from there following packages:

“bs” with sub-packages “http” and “plaintext”

“pd” – it’s updated form of the Location.java